BAB II

TINJAUAN PUSTAKA

2.1 Sekolah Menengah Atas Negeri 2 Purwakarta

Sekolah Menengah Atas Negeri 2 Purwakarta berdiri pada tanggal 3 September 1979 berdasarkan SK pendirian Menteri Pendidikan dan Kebudayaan RI dengan No Statistik Sekolah (NISS) :3010220001002, dalam kurun waktu 29 tahun SMAN 2 Purwakarta telah dipimin oleh 11 orang Kepala Sekolah, yaitu

1. Bpk. M. Masri (Alm),
2. Bpk DRS. Akhmad Subarkah,
3. Bpk Drs. H. Machmud Radi,
4. Bpk. H. Waryo Sukanda, BA
5. Ibu Dra. Hj. R. Atty Ruchyati
6. Bpk. Drs. Iri Setiadi,
7. Bpk Drs. Jumdiat Marzuki, MM
8. Ibu Dra. Iis Sri Sugiharti
9. Bpk. Drs Asep Saepudin
10. Plt. Kepala Bpk. Drs. Rudiana, MM
11. BPK. Drs. Marseno
12. dan sekarang dipimpin oleh BPK. Drs. H. Gaos Suherman, M.Pd

SMAN 2 Purwakarta memiliki tujuan, yaitu dalam Mewujudkan iklim belajar yang kondusif, selalu berusaha melengkapi sarana prasarana khususnya sarana prasarana

pengembangan ruangan belajar,meningkatkan partisipasi orang tua untuk terlaksananya program sekolah, melaksanakan pembelajaran team teaching untuk meningkatkan mutu pembelajaran, membekali keterampilan komputer bagi siswa untuk menunjang mata pelajaran dan mengahadapi perkemabangan IPTEK, dan Mengembangkan pembelajaran yang berorientasi kepada KTSP (Kurikulum Tingkat Satuan Pendidikan).

Adapun Visi dan Misi SMA Negeri 2 Purwakarta, Visi SMA Negeri 2 Purwakarta :

1. Membangun Generasi Kreatif, Inovatif dan Tangguh yang dilandasi oleh keimanan dan ketaqwaan.

Misi SMA Negeri 2 Purwakarta :

1. Meningkatkan kepribadian dan kemadirian yang dilandasi iman dan taqwa.
2. Meningkatkan prestasi akademik dan non akademik agar tercapai mutu lulusan yang berkualitas.
3. Meningkatkan profesionalisme Guru dan TU, membina semangat kerjasama yang dilandasi kekeluargaan.
4. Menciptakan suasanan yang kondusif bagi terwujudnya harmonisasi civitas sekolah.
5. Meningkatkan Wawasan Wiyata Mandala untuk mendukung kegiatan belajar mengajar

Tujuan SMA Negeri 2 Purwakarta :

1. Mewujudkan iklim belajar yang kondusif.

2. Melengkapi srana prasarana khususnya sarana prasarana pengembangan ruangan belajar.

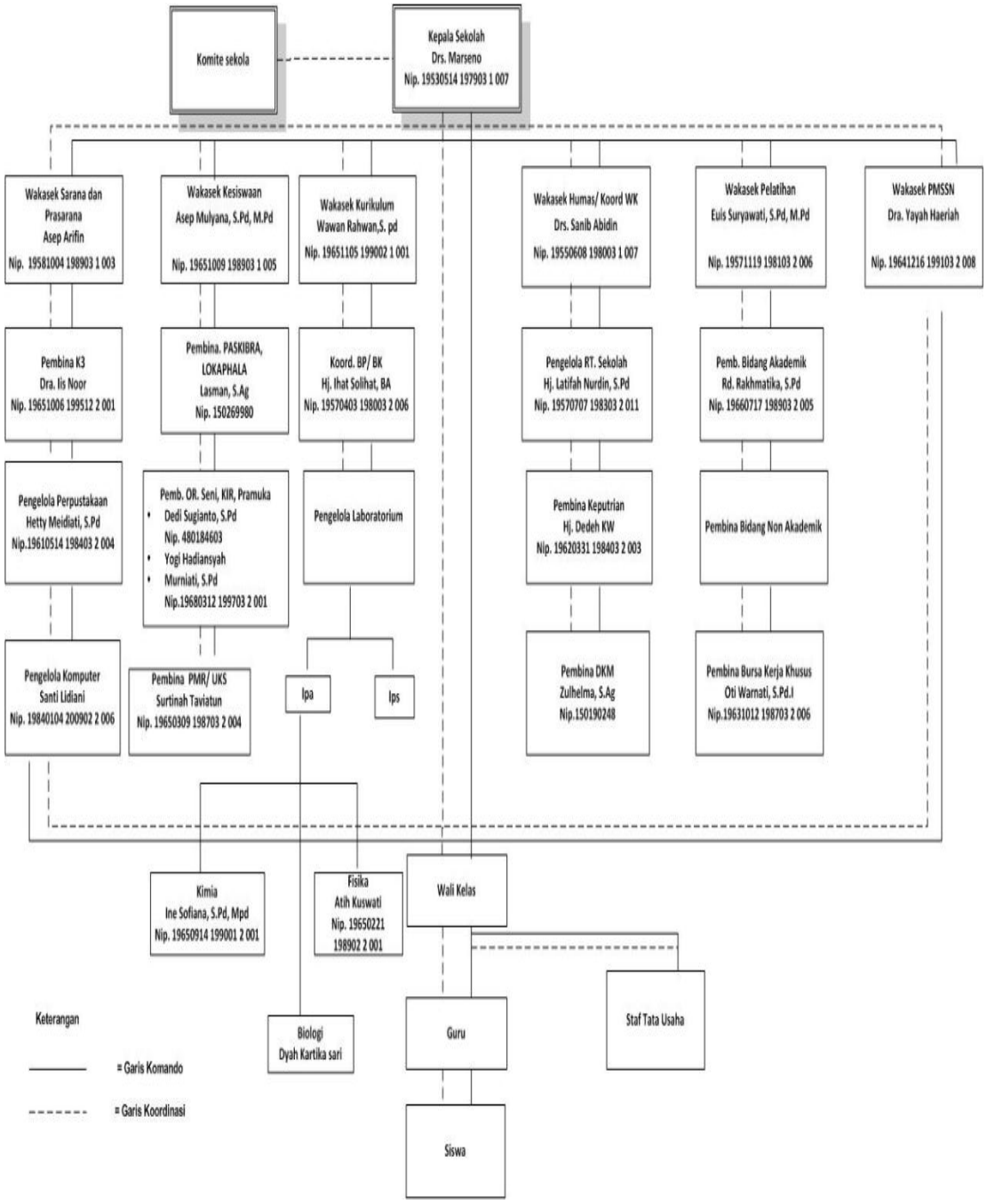
3. Meningkatkan partisipasi orang tua untuk terlaksananya program sekolah.

4. Melaksanakan pembelajaran team teaching untuk meningkatkan mutu pembelajaran.

5. Membekali keterampilan komputer bagi siswa untuk menunjang mata pelajaran dan mengahadapi perkemabangan IPTEK.

6. Mengembangkan pembelajaran yang berorientasi KTSP (kurikulum 2006) dan BBE Life Skill untuk pengembangan jiwa mandiri siswa.

2.1.1. Struktur Organisasi SMA N 2 Purwakarta



Gambar 2. 1 Struktur Organisasi SMA N 2 Purwakarta

PEMBAGIAN TUGAS KEPSEK, WAKASEK , PKS, PJP, KOORDINATOR, PEMBINA , DAN WALI KELAS TAHUN PELAJARAN 2011 – 2012

A. KEPSEK, WAKASEK DAN PEMBINA PROFESI

No	Nama / NIP	Pangkat / Golongan	Tugas
1.	Drs. H. Gaos Suherman, M.Pd 10560619 198103 1 012	Pembina Tk.I (IV-b)	Kepala Sekolah
2.	Drs.H. Sanib Abidin 19550608 198003 1 007	Pembina TK. I (IV/b)	Wakasek Umum / Koordinator
3.	Wawan Rahwan, S.Pd. 19651105 199002 1 001	Pembina (IV/a)	Wakasek. Urusan. Kurikulum
4.	Samsuri, S.Pd 19630505 198803 1 011	Pembina (IV/a)	Wakasek Urusan. Kesiswaan
5.	Asep Arifin 19581004 198903 1 003	Pembina (IV/a)	Wakasek Urusan Sarana Prasarana
6.	Euis Suryawati, M.Pd 19630912 198903 1 015	Pembina (IV/b)	Wakasek Humas
7.	Dra. Yayah Haeriah 19641216 199103 2 008	Pembina (IV/a)	Wakasek Urusan PM SSN & Personil
8.	Dra.Hj. Euis Suwartini Y,M.M 19600929 198403 2 004	Pembina (IV/a)	Wakasek Urusan Pelatihan/Diklat
9.	Drs. Anwar Khusnansyah 19511215 198003 1 005	GTT	Pembina Profesi Guru

B. PKS

No	Nama / NIP	Pangkat / Golongan	Tugas
1.	H O N O, S.Pd 19630912 198903 1 015	Pembina (IV/a)	PKS Urusan Kurikulum
2.	Reni Nuraeni, S.Pd. 19620702 198512 2 001	Pembina (IV/a)	PKS Urusan Kesiswaan /Pembina OSIS

3.	Drs. Tikno Slamet 19591014 198709 1 001	Pembina (IV/a)	PKS Urusan Kesiswaan /Pembina Non OSIS
4.	Lasman, S.Ag 150 269 980	Penata Tk.I (III-d)	PKS Urusan Sarana dan Prasarana
5.	Hj.Latifah Nudin, S.Pd 19570707 198303 2 011	Pembina (IV/a)	PKS Urusan Humas
6.	Kasidi, S.Pd.Fis 19690410 199201 1 001	Pembina (IV/a)	PKS Urusan PM SSN
7.	Dra. Iis Aisyah 19600620 198803 2 003	Pembina (IV/b)	PKS Urusan Pelatihan

C. PEMBAGIAN TUGAS PENANGGUNG JAWAB PROGRAM (PJP),KOORDINATOR,DAN , PEMBINA DIBAWAH KOORDINASI WAKASEK

1. Penanggung Jawab Program (PJP)

No	Nama / NIP / NUPTK	Pangkat / Golongan	Tugas
1.	Mokh. Adnan, S.Sos 8354 7536 5620 0003	-	PJP Guru Piket
2.	A N I S A, S.Pd 19790209 200801 2 001	Penata Muda (III-a)	PJP Lab.Biologi
3.	Drs. Tarsudin 19581111 198403 1 008	Pembina (IV/a)	PJP Lab.Kimia
4.	Atih Kuswati 19650221 198902 2 001	Pembina (IV/a)	PJP Lab.Fisika
5.	R. Hasanah, S.Pd. 19540606 198003 2 007	Pembina (IV/a)	PJP Lab.Bahasa
6.	Dida Nurhuda, A.Md 5433 7596 6020 0062	-	PJP Lab.Komputer
7.	Heti Meidiati, S.Pd 19610514 198403 2 004	Pembina (IV/a)	PJP Perpustakaan
8.	Nandang, S.Ag 0834 7476 5120 0022	-	PJP Baca Al-Qur'an dan Tafsir

2. Koordinator BP/BK

No	Nama / NIP /NUPTK	Pangkat / Gol	Tugas
1.	Oti Warnati, S.Pd	Pembina (IV-a)	Koordinator BP/BK

	19631012 198703 2 006		

3. Pembina

No	Nama / NIP /NUPTK	Pangkat / Gol	Tugas
1.	Rd. Rubbin Kalagan, S.Pd 7163 7606 6220 0013	GTT	Pembina Paskibra
2.	Dra. Tien Martini 19630516 198305 2 004	Pembina (IV/a)	Pembina PMR / UKS dan Seksi 7 OSIS
3.	Zulhelma, S.Ag 150 190 248	Pembina (IV/a)	Pembina DKM & Seksi 1 OSIS
4.	Hj. Dedeh Komariah W, S.Pd.I 19620331 198403 2 003	Pembina (IV/a)	Pembina BKK
5.	Drs. Ade Surachman 9844 7446 4720 0042	GTT	Pembina Pramuka Putra
6.	Surtinah Taviatun, S.Pd. 19650309 198703 2 004	Pembina (IV/a)	Pembina Pramuka Putri & Seksi 3 OSIS
7.	Masriyah, S.Ag 19710305 200801 2 004	Penata Muda (III/a)	Pembina Keputrian &Seksi 2 OSIS
8.	Dedi Sugianto, S.Pd. 19780829 200801 1 002	Penata Muda (III/a)	Pembina Olahraga & Seksi 4 OSIS
9.	Anggie Nugeraha, S.Sn	GTT	Pembina Seni
10.	Rd. Rakhmatika S.H., S.Pd. 19660717 198903 2 005	Pembina (IV/a)	Pembina Olimpiade Mata Pelajaran
11.	Dra. Iis Noor Rosidah 19651006 199512 2 001	Pembina (IV/a)	Pembina K3
12.	Iis Mulyawati, S.Pd 19700708 200701 2 004	PenataMudaTk.I (III/b)	Pembina Majalah Dinding & Majalah Sekolah & Seksi 8 OSIS
13.	Santi Lidiani, ST 19840401 200902 2 006	Pembina (IV/a)	Pembina KIR/Sains dan Teknologi
14.	Dra. Siti Jufriyah 19620630 200604 2 001	PenataMudaTk.I (III/b)	Pembina Koperasi Siswa dan Seksi 6 OSIS
15.	Dyah Kartikasari, S.Pd 3644762663300062	GTT	Pembina Pencinta Alam / Adiwiyata & Seksi 5 Osis
16.	Mokh. Adnan , S,Sos 8354 7536 5620 0003	GTT	Pembina Klub Komputer dan Seksi 9 OSIS

17.	Yayan Sopian, S.Pd 6839 7606 6120 0030	-	Pembina Klub Bahasa Inggris dan Seksi 10 OSIS

2.2 Penjadwalan

Berdasarkan Kamus Besar Bahasa Indonesia menjelaskan bahwa:

ñ Jadwal adalah daftar atau tabel rencana kegiatan dengan pembagian waktu ò.

Penjadwalan adalah proses pembuatan susunan kegiatan dengan pembagian waktu pelaksanaan, Jadi dapat disimpulkan bahwa sistem informasi penjadwalan adalah suatu cara didalam suatu organisasi untuk menghasilkan suatu informasi penjadwalan yang lebih berguna bagi para pengguna. Atau juga sistem informasi penjadwalan dapat diartikan sebagai sistem yang memberikan layanan informasi yang berupa data yang berhubungan dengan proses pembuatan susunan kegiatan dengan pembagian waktu pelaksanaan dalam sebuah organisasi atau instansi.

Penjadwalan dapat didefinisikan sebagai pengaturan pengalokasian sumber daya dalam jangka waktu tertentu untuk melakukan serangkaian tugas yang telah dirancangkan, penjadwalan adalah aturan atau proses pengorganisasian, pemilihan, dan penentuan waktu penggunaan tempat atau sumber-sumber untuk mengerjakan semua aktivitas yang diperlukan yang memenuhi kendala aktivitas dan sumber daya.

2.3 Algoritma Dalam Penjadwalan

a) Algoritma Semut

Perilaku sekumpulan semut dalam mencari atau menemukan sumber makanan dari sarangnya yang kemudian kembali lagi kesarangnya, merupakan inspirator ditemukannya algoritma semut[10]. Kemampuannya dalam menemukan solusi yang baik dari sekian solusi yang ada dikarenakan penggunaan informasi lokal berupa pheromone yang ditinggalkan setiap kali semut melewati suatu jalur. Semakin banyak semut yang melalui jalur tersebut maka pheromone yang ditinggalkan semakin banyak sehingga semut untuk menemukan jalur selanjutnya dengan menggunakan pemilihan acak yang disesuikan berdasarkan besarnya pheromone yang dimiliki oleh jalur tersebut. Implementasi algoritma semut telah banyak dilakukan seperti pada permasalahan TSP, Spanning tree dan penjadwalan. Pada permasalahan penjadwalan produksi jobshop, dalam penerapannya diperlukan teknik pendekatan graph yang akan menjadikan sebuah representasi dari urutan joboperasi dari kasus penjadwalan yang diberikan.

Ada tiga jenis algoritma semut yaitu : ant cycle, ant-density dan ant-quantity. membedakan antara ketiga jenis algoritma semut tersebut adalah pada proses pembarharuan pheromone semut. Pada penelitian ini pendekatan algoritma semut yang digunakan adalah ant cycle untuk menemukan waktu penyelesaian keseluruhan job yang paling optimal.

b) Algoritma Genetika

Algoritma genetika adalah suatu algoritma pencarian yang meniru mekanisme dari genetika alam. Algoritma Genetika banyak dipakai pada aplikasi bisnis, teknik maupun pada bidang keilmuan lainnya. Algoritma ini dimulai dengan kumpulan solusi yang disebut dengan populasi[9]. Solusi-solusi dari sebuah populasi diambil dan digunakan untuk membentuk populasi yang baru. Hal ini dimotivasi dengan harapan bahwa populasi yang baru dibentuk tersebut akan lebih baik daripada yang lama. Solusi-solusi yang dipilih untuk membentuk solusisolusi yang baru dipilih sesuai dengan fitness mereka masing-masing.

Di dalam proses tersebut dapat terjadi variasi yang disebabkan karena adanya mutasi, sehingga keturunan yang dihasilkan dapat mempunyai kelebihan bahkan tidak memiliki kekurangan dari orangtuanya. Setiap makhluk hidup akan mengalami seleksi alam, sehingga makhluk hidup yang mempunyai kemampuan untuk beradaptasi dengan lingkungan sekitarnya dapat bertahan sampai generasi selanjutnya. Semakin bagus atau sesuai fitness dari sebuah solusi maka solusi tersebut mempunyai peluang besar untuk dipilih. Proses ini dilakukan berulang sampai kondisi tertentu dipenuhi.

c) Algoritma Backtrack

Backtrack adalah algoritma berbasis DFS untuk mencari solusi persoalan. Backtracking merupakan algoritma perbaikan dari *brute-force*, yang secara sistematis hanya mencari solusi yang mungkin. Umumnya algoritma *Backtrack* bersifat rekursif, namun ada pula versi *backtrack* yang iteratif [5].

Algoritma runut balik pertama kali diperkenalkan oleh D.H Lehmer pada tahun 1950. Algoritma ini cukup mangkus untuk digunakan dalam beberapa penyelesaian masalah dan juga untuk memberikan kecerdasan buatan dalam *game*. Beberapa *game* populer semisal Sudoku, Labirin, Catur juga bisa diimplementasikan dengan menggunakan algoritma runut balik.

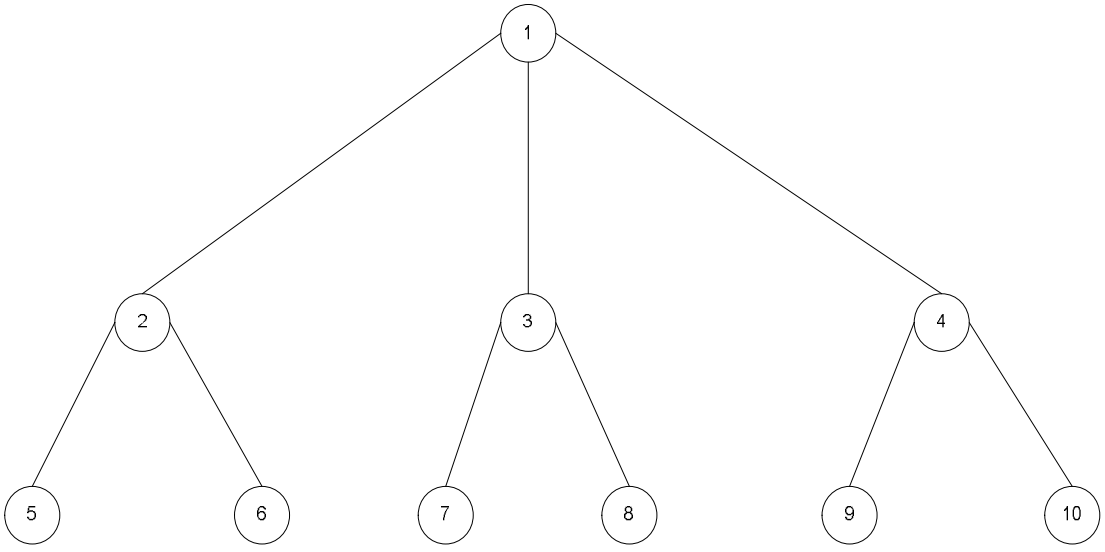
Algoritma runut balik (*backtrack*) merupakan algoritma yang digunakan untuk mencari solusi persoalan secara lebih mangkus dari pada menggunakan algoritma *brute force.* Algoritma ini akan mencari solusi berdasarkan ruang solusi yang ada secara sistematis namun tidak semua ruang solusi akan diperiksa, hanya pencarian yang mengarah kepada solusi yang akan diproses.

Algoritma runut balik berbasis pada DFS (*Depth First Search*) sehingga aturan pencariannya akan mengikut kepada aturan pencarian DFS yaitu dengan mencari solusi dari akar ke daun (dalam pohon ruang solusi) dengan pencarian mendalam. Simpul-simpul yang sudah dilahirkan (diperiksa) dinamakan simpul hidup (*live node*). Simpul hidup yang sedang diperluas dinamakan simpul-E atau *Expand Node.*

Depth-first search atau DFS adalah sebuah algoritma untuk melakukan penelusuran atau pencarian pada pohon, struktur pohon, atau graf. Penelusuran atau pencarian ini dimulai pada simpul akar dan menelusuri sejauh mungkin sepanjang cabang yang ada. *Depth-first search* ini pertama kali ditemukan pada abad ke-19 oleh ahli matematika dari Prancis bernama Charles Pierre Trémaux. Secara formal, DFS adalah sebuah pencarian kurang terinformasi (*uninformed search)* yang berkembang

dengan penelusuran yang dimulai dari simpul anak pertama sebuah pohon dan terus masuk hingga simpul yang diinginkan ditemukan, atau hingga mencapai simpul yang tidak mempunyai anak. Setelah itu, terjadi *backtracking* sehingga penelusurannya kembali ke simpul terdekat yang belum dikunjungi. Biasanya, pada implementasi yang tidak rekursif, semua simpul yang telah ditelusuri dengan DFS ini dimasukkan ke dalam sebuah *stack.*

Algoritma *backtrack* mempunyai prinsip dasar yang sama seperti *brute-force* yaitu mencoba segala kemungkinan solusi. Perbedaan utamanya adalah pada ide dasarnya, semua solusi dibuat dalam bentuk pohon solusi (pohon ini tentunya berbentuk abstrak) dan algoritma akan menelusuri pohon tersebut secara *DFS* (*depth field search)* sampai ditemukan solusi yang layak. Agar lebih jelas bisa dilihat pada pohon solusi berikut:



Gambar 2.1 Pohon solusi

Keterangan Gambar Pohon Solusi :

Misalkan pohon diatas menggambarkan solusi dari suatu permasalahan. Untuk mecapai solusi (5), maka jalan yang ditempuh adalah (1,2,5), demikian juga dengan solusi-solusi yang lain. Algoritma backtrack akan memeriksa mulai dari solusi yang pertama yaitu solusi (5). Jika ternyata solusi (5) bukan solusi yang layak maka algoritma akan melanjutkan ke

solusi (6). Jalan yang ditempuh ke solusi (5) adalah (1,2,5) dan jalan untuk ke solusi (6) adalah (1,2,6). Kedua solusi ini memiliki jalan awal yang sama yaitu (1,2). Jadi daripada memeriksa ulang dari (1) kemudian (2) maka hasil (1,2) disimpan dan langsung memeriksa solusi (6). Pada pohon yang lebih rumit, cara ini akan jauh lebih efisien daripada brute-force.

2.4 Alat Bantu Pemodelan

2.4.1 Perancangan Berorientasi Objek

Rekayasa adalah bentuk upaya manusia untuk mencari atau merancang produk yang berkualitas. Produk dapat berupa sistem atau perangkat lunak yang sedang atau akan dibangun dengan satu model pengembangan. Orientasi objek adalah cara pandang algoritma yang di implementasikan pada bahasa pemrograman berorientasi objek.

OOA adalah teknik yang model driven yang mengintegrasikan data dan proses yang disebut objek. Model OOA adalah gambar-gambar yang mengilustrasikan objek-objek sistem dari berbagai perspektif, seperti struktur, perilaku dan interasi

antar objek. Contoh yang paling terkenal adalah UML (Unified Modelling Language).

Cara berpikir orienasi objek adalah segala sesuatu dipandang dengan objek. Objek dapat berupa konsep, abstraksi, atau sesuatu dengan batas-batas tegas dan mempunyai arti untuk persoalan yang ditangani. Objek mempunyai identitas dan dapat dibedakan. Ada dua kegunaan objek yaitu : untuk meningkatkan pemahaman dunia nyata dan menyediakan dasar pengetahuan praktis untuk implementasi komputer.

Perangkat lunak masih seperti model yang digunakan dalam metodologi konvensional, yaitu dapat menggunakan model *waterfall*. Dalam model *waterfall* terdapat lima tahapan proses, yaitu : tahap rekayasa sistem, analisis perancangan, pemrograman, pengujian dan pemeliharaan. Dengan menggunakan *waterfall*, memungkinkan adanya pengembangan aplikasi dikemudian hari, sehingga aplikasi yang dibangun dapat bertahan dalam jangka waktu yang panjang.

2.4.2 UML (*Unified Modeling Language*)

UML Singkatan *dari Unified Modeling Language* yang berarti bahasa pemodelan standar [3]. mengatakan sebagai bahasa, berarti *UML* memiliki sintak dan semantic. Ketika kita membuat model menggunakan konsep *UML* ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi?

Bagaimana keamanan terhadap sistem yang kita buat? Dan sebagainya dapat dijawab dengan *UML*. UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML telah diaplikasikan dalam bidang investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales dan supplier.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis timing diagram) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensuport para pengembang sistem saat ini. Sebagai perancangan sistem, maua tidak mau pasti akan menjumapai *UML,* baik kita sendiri yang membuat atau sekedar membaca diagram *UML* buatan orang lain [4]

2.4.2.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah ñapaò yang diperbuat sistem, dan bukan ñbagaimanaò. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah use case dapat meng-*include* fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di-*include* akan dipanggil setiap kali use case yang meng-*include* dieksekusi secara normal. Sebuah use case dapat di-*include* oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common* [2].

2.4.2.2 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi) [2].

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

2.4.2.3 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum [2].

2.4.2.4 Sequence Diagram

Sequence Diagram adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada

sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa òpesan/*message*ò.

Sequence Diagram digunakan untuk menggambarkan skrenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian untuk menghasilkan *output* tertentu. Sequence. Diagram diawali dari apa yang me-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan [2].

2.4.2.5 Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya, komponen piranti lunak adalah modul berisi code, baik berisi source code maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time, link time*, maupun *run time*.

Umumnya komponen terbentuk dari beberapa *class* atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain [2].

2.4.2.6 Deployment Diagram

Deployment/physical diagram menggambarkan *detail* bagaimana komponen di-*deploy* dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal. Sebuah node adalah server,

workstation, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini [2].

2.5 Pengertian Sitem Informasi

Informasi merupakan salah satu sumber daya penting dalam suatu organisasi; digunakan sebagai bahan pengambilan keputusan. Sehubungan dengan hal itu, informasi haruslah berkualitas [6]. Kualitas informasi ditentukan oleh tiga faktor, yaitu :

1. Relevasi,
2. Tepat waktu, dan
3. Akurasi.

Sistem informasi dapat dikatakan sebagai suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam suatu sistem organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Kegiatan sistem informasi mencakup :

1. *Input*, menggambarkan kegiatan untuk menyediakan data untu diproses.
2. Proses, menggambarkan bagaimana suatu data diproses untuk mengasilkan suatu informasi yang bernilai tambah.
3. *Output*, suatu kegiatan untuk menghasilkan laporan proses diatas tersebut.
4. Penyimpanan, suatu kegiatan untuk memelihara dan menyajikan data.
5. Kontrol, suatu aktifitas untuk menjamin bahwa sistem informasi tersebut sesuai yang diharapkan.

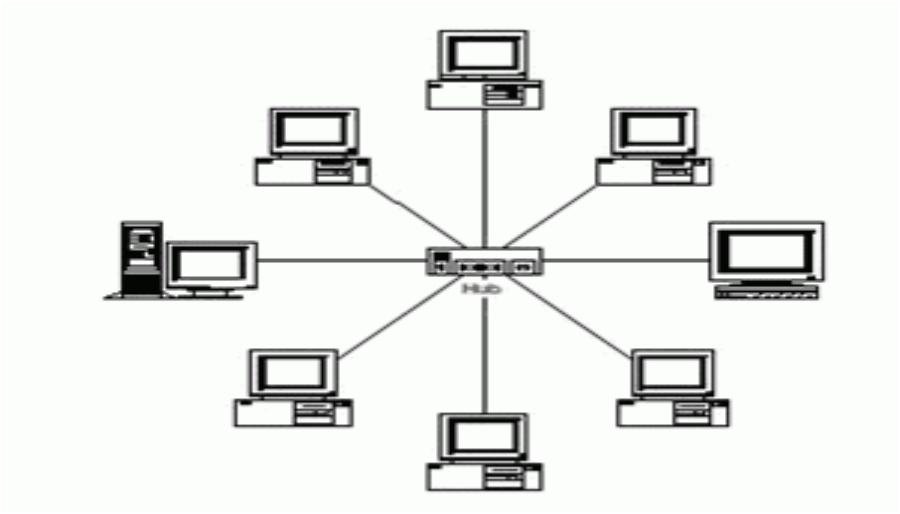
2.6 Jaringan yang ada di SMAN 2 Purwakarta

Client / Server merupakan sebuah paradigma dalam teknologi informasi yang merujuk kepada cara untuk mendistribusikan aplikasi ke dalam dua pihak yaitu *client* dan *server*. *Server* adalah komputer yang dapat *member*ikan *service* ke *client*, sedangkan *client* adalah komputer yang mengakses beberapa *service* yang ada di *server*. Ketika *client* membutuhkan suatu *service* yang ada di *server*, dia akan mengirim *request* kepada *server* lewat jaringan. Jika *request* tersebut dapat dilaksanakan, maka *server* akan mengirim balasan berupa *service* yang dibutuhkan untuk saling berhubungan menggunakan *Socket*.

1. Karakteristik *Server*
    a. Pasif
    b. Menunggu *request*
    c. Menerima *request*, memproses mereka dan mengirimkan balasan berupa *service*
2. Karakteristik *Client*
    a. Aktif
    b. Mengirim *request*
    c. Menunggu dan menerima balasan dari *server*

Socket adalah sebuah endpoint untuk komunikasi didalam jaringan. Sepasang proses atau thread berkomunikasi dengan membangun sepasang socket, yang masing-masing proses memilikinya. *Socket* dibuat dengan menyambungkan dua buah alamat IP melalui *port* tertentu. Secara umum *socket* digunakan dalam *client/server* sistem,

dimana sebuah *server* akan menunggu *client* pada *port* tertentu. Begitu ada *client* yang menghubungi *server* maka *server* akan menyetujui komunikasi dengan *client* melalui *socket* yang dibangun.



Gambar 2. 2 Jaringan yang ada di SMAN 2 Purwakarta

2.7 Java

Java adalah bahasa pemrograman serbaguna. Java dapat digunakan untuk membuat suatu program sebagaimana anda membuatnya dengan bahasa seperti pascal atau C++. Yang lebih menarik, java juga mendukung sumber daya Internetyang saat ini popular, yaitu World Wide Web atau yang sering disebut Web saja. Java juga mendukung aplikasi klien/server, baik dalam jaringan local (LAN) maupun jaringan beskala luas (WAN).

Dalam sejumalah literature disebutkan bahwa Java merupakan hasil perpaduan sifat dari sejumlah bahasa pemograman, yaitu C, C++, Objek-C, SmallTalk, dan Common LISP. Selain itu Java juga dilengkapi dengan unsure

keamanan. Yang tak kalah penting adalah bahwa Java menambahkan paradigma pemograman yang sederhana. Jika Anda telah mengenal C atau C++, yang mengandalkan pointer dan Anda dapat merasakan keruwetannya, Java justru meninggalkannya sehingga Anda akan memperoleh kemudahan saat menggunakannya [8].

2.8 My SQL

MySQL adalah multi user database yang menggunakan bahasa *Strucktured Query Language* (*SQL*). *MySQL* merupakan *database server multi user* dan *multi threaded* yang tangguh. Dengan memiliki banyak *feature MySQL* bisa bersaing dengan *database* komersial sekalipun. Tidak mengejutkan *MySQL* menjadi database pilihan untuk banyak pengguna Delphi 6. *MySQL* termasuk jenis *RDBMS* (*Relational Database Management System*) itulah sebabnya istilah seperti tabel, baris dan kolom digunakan pada *MySQL*. Pada *MySQL* sebuah *database* mengandung satu atau sejumlah tabel. Tabel terdiri dari sejumlah baris dan setiap baris mengandung satu atau beberapa kolom[7].

MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan *MySQL* yaitu TcX, mengaku menyimpan data lebih dari 40 *database*, 10-000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 *Gigabyte* data. SQL adalah bahasa standar yang digunakan untuk mengakses *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi 25 dan digunakan sebagai standar industri.

Dengan menggunakan SQL, proses akses *database* menjadi lebih *user* ï *friendly* dibandingkan dengan menggunakan *DATABASE* atau *Clipper* yang masih menggunakan perintah-perintah pemrograman. *MySQL* merupakan *software database* yang paling populer di lingkungan *Linux*, kepopuleran ini karena ditunjang performa *query* dari databasenya yang saat ini bisa dikatakan paling cepat dan jarang bermasalah. *MySQL* ini juga sudah dapat berjalan pada lingkungan *Windows*.

Beberapa karakteristik utama dari *MySQL* :

1. Fully multi-threaded dengan kernel threads. Artinya bisa dengan mudah mempergunakan multiple CPU bila ada.
2. API (Application Programming Interface) dengan C, C++, Delphi, Eiffel, Java, Perl, PHP, VB 6, Python dan Tcl.
3. Beroperasi pada banyak platform berbeda seperti : Linux, Win 95, Win 98, NT, Win 2000, Solaris, SunOS dll.
4. Kita bisa mencampurkan tabel dari database yang berbeda pada query yang sama.
5. Dukungan ODBC (Open-Database-Connectivity), misalkan dengan MS Access bisa melakukan koneksi ke MySQL.
6. Server bisa memberikan pesan kesalahan ke client dalam banyak bahasa.